



TAMPERE UNIVERSITY OF TECHNOLOGY

DIEGO PINEDO ESCRIBANO
ANÁLISIS DEL DESARROLLO DE APLICACIONES MÓVILES
MULTI PLATAFORMA

Versión resumida en Español

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

PINEDO ESCRIBANO, DIEGO: Análisis del desarrollo de aplicaciones móviles multiplataforma.

Major subject: Software Engineering

Tutor: Professor Tommi Mikkonen

Co-tutor: Francisco Javier García Blas

Grade: 3

Grade System:

1 – Satisfactory. 2 – Very satisfactory. 3 – Good. 4 – Very good. 5 - Excellent

Keywords: aplicaciones móviles, multiplataforma, Android, iOS, Appcelerator Titanium, aplicación nativa, aplicación web

El desarrollo de aplicaciones móviles es un mercado muy grande en estos tiempos que corren. Existen muchas compañías que están invirtiendo mucho dinero para desarrollar aplicaciones exitosas y rentables. El problema surge cuando las empresas quieren desarrollar aplicaciones para ser usadas por el mayor número de personas posibles. Hoy en día existen múltiples dispositivos en el mercado (Android, iOS, BlackBerry OS, Windows Phone, etc) que hacen más difícil esa tarea. Por esta razón en los últimos años han surgido diferentes tecnologías que ayudan al desarrollo de aplicaciones multiplataforma.

En este proyecto una de estas tecnologías es analizada. Durante el proceso de análisis se explica qué es una aplicación móvil. Además se explica qué es una aplicación móvil multiplataforma especificando sus características. Finalmente se introduce al lector a la tecnología Appcelerator Titanium explicando las características de esta herramienta.

Con el fin de mostrar los resultados de una manera más visual, durante el análisis se desarrolla una aplicación móvil multiplataforma de prueba que es instalada nativamente en un dispositivo Android y otro iOS. Esta aplicación contiene algunas de las características más importantes que toda aplicación móvil debería tener.

Como punto final y a modo de conclusiones de este proyecto se puede extraer que las aplicaciones móviles multiplataforma tienen algunas ventajas comparadas con las aplicaciones móviles tradicionales, pero el nivel de madurez en los procesos de desarrollo y en las herramientas usadas para su creación son todavía demasiado bajos.

TABLE OF CONTENTS

1.	Introducción	3
2.	Aplicaciones Móviles.....	4
2.1.	Tipos de aplicaciones móviles.....	4
2.2.	Componentes.....	5
2.2.1.	Interfaz Gráfica de Usuario.....	5
2.2.2.	Lógica de la aplicación e interfaces.....	5
2.3.	Cross-platform application.....	6
3.	Appcelerator Titanium	7
4.	Ánalysis de una aplicación móvil multi plataforma	8
4.1.	Caso de uso: Aplicación para crear guías de viaje	8
4.2.	Análisis de los requisitos.....	9
4.3.	Diseño de la aplicación	11
4.3.1.	Módulo UI.....	13
4.3.2.	Módulo lib.....	13
4.3.3.	Módulo App	13
4.3.4.	Pantallazos	14
4.4.	Evaluación del framework	18
5.	Conclusions	21

1. INTRODUCCIÓN

El desarrollo de aplicaciones móviles es un mercado enorme en la actualidad. Hay muchas empresas que invierten mucho dinero para desarrollar exitosas y rentables aplicaciones. El problema aumentará cuando se trata de desarrollar una aplicación para ser utilizado por todos los usuarios independientemente de la plataforma que estén usando (Android, iOS, BlackBerry OS, Windows Phone, etc.) La forma tradicional de hacer frente a este problema ha sido siempre la de contratar a más expertos y dividir el proceso de desarrollo en las corrientes diferentes, cada uno cuidando de un objetivo. Sin embargo, estas soluciones pueden parecer ineficiente para una pequeña empresa que no puede permitirse contratar a un experto. Por esta razón, en los últimos años muchas soluciones diferentes han aparecido que hacen que el desarrollo de aplicaciones multiplataforma más fácil. Estas soluciones se basan generalmente en un marco con herramientas específicas que le permite al usuario programar en una tecnología middleware, y como resultado se puede obtener una aplicación que se ejecuta en diferentes plataformas (los que la solución específica permite desarrollar).

El objetivo de esta tesis es analizar aplicaciones multiplataforma móviles y, en concreto, los producidos utilizando Titanium Appcelerator. Con el fin de lograr este objetivo, Appcelerator serán analizado y otros marcos serán revisados. Durante el proceso de análisis vamos a explicar lo que es una aplicación móvil, estableciendo la diferencia entre los dos tipos existentes de las aplicaciones, nativas de aplicaciones y aplicaciones web. También se explicará lo que es una aplicación móvil multi plataforma y cuáles son sus características. Por último, Se introduce Titanium Appcelerator, presentando las características que esta tecnología ofrece, el análisis de la homogeneidad de las aplicaciones producidas por el marco, sus problemas y sus posibles soluciones. Con el fin de mostrar los resultados de una forma más visual, una aplicación multi-plataforma móvil que se instalará de forma nativa en Android y plataformas iOS se desarrollará. Contendrá todas las características más importantes que una aplicación de telefonía móvil debe tener.

Esta tesis se divide en cuatro capítulos. En el capítulo 1 hay una introducción a la tesis. En el capítulo 2 se explica lo que es una aplicación móvil, los tipos de aplicaciones y después de eso nos vamos a centrar en lo que es una aplicación multi-plataforma móvil. En el capítulo 3 se analiza el marco Titanium Appcelerator. El capítulo 4 se explica cómo desarrollar una aplicación multiplataforma con Appcelerator. Una aplicación móvil se desarrollará. Por último, el capítulo 5 se exponen las conclusiones de esta tesis.

2. APLICACIONES MÓVILES

Una aplicación móvil de mano es un software desarrollado para ser utilizado en cualquier tipo de dispositivos móviles como teléfonos inteligentes o tabletas. Este tipo de aplicaciones pueden ser aplicaciones independientes o pueden ser aplicaciones distribuidas repartidos por diferentes dispositivos, redes y / o servidores. Hoy en día casi todas las aplicaciones están desarrolladas para teléfonos inteligentes y tabletas que ofrecen herramientas muy buenas y de gran alcance y los marcos para el desarrollo de este software.

2.1. Tipos de aplicaciones móviles

Hay dos tipos de aplicaciones de telefonía móvil: las aplicaciones nativas y aplicaciones basadas en tecnologías web. Una aplicación nativa es el que fue diseñado para ser instalado en un sistema operativo específico (iOS, Android, Symbian, etc) y también es dependiendo del firmware del dispositivo. Con el fin de tener una aplicación que se ejecuta en los cambios de diferentes dispositivos o modelos pequeños o grandes serán necesarios. Aplicaciones basadas en web son desahogado el uso de tecnologías web como HTML o JavaScript, y deberá ser interpretado por los navegadores web de los diferentes sistemas operativos.

Hay varias diferencias entre las aplicaciones nativas y la web. Cada uno de ellos tiene sus propias ventajas y desventajas. Con base en los resultados obtenidos por el Global Intelligence Alliance [1] y el análisis realizado por Worklight [2], es posible extraer los puntos clave que marcan la diferencia entre las aplicaciones nativas y aplicaciones web. El primer punto es la experiencia del usuario, este es uno de los factores más importantes a considerar al decidir qué tipo de aplicación se va a desarrollar. Los resultados extraídos del estudio muestra que las aplicaciones nativas tienen la capacidad de construir una interfaz de usuario de Superior, por lo tanto la experiencia del usuario es mejor en las aplicaciones nativas que con aplicaciones basadas en web.

La otra diferencia importante que destacó la investigación es que el acceso a las principales características del dispositivo es aún mejor en aplicaciones nativas, pero la evolución de las tecnologías, en particular, HTML5, es equilibrar la balanza con las aplicaciones Web. Se trata de una cuestión de tiempo que las tecnologías Web tienen acceso a la cámara del dispositivo, el GPS, contactos, calendario, sistema de archivos, el acelerómetro, y así sucesivamente.

La siguiente característica comentó por la Global Intelligence Aliance es que con las aplicaciones Web que los desarrolladores tienen un control total sobre la aplicación y tienen un control directo sobre la distribución propia, sin necesidad de buscar la aprobación de 3^a parte del proveedor. En algunos casos esto es una ventaja, ya que es posible obtener ganancias de la venta total o anuncios dentro de la aplicación. Por otro lado, tercera de otros fabricantes, como la App Store de Apple o el Android Market de Google proporcionan una poderosa herramienta de marketing para promocionar y distribuir las aplicaciones.

Para resumir la información extraída de ambos estudios, en la figura 2.1 se presenta una comparación directa entre las aplicaciones basadas en web y nativas.



Figure 2.1 Comparison between Native and Web apps from Worklight [2]

2.2. Componentes

Una aplicación móvil puede ser dividido en la interfaz gráfica de usuario (GUI), y la lógica de la aplicación y las interfaces con los periféricos que tiene.

2.2.1. Interfaz Gráfica de Usuario

El interfaz gráfico de usuario es el punto de acceso donde el usuario controlar el comportamiento de la aplicación. Incluye todos los elementos gráficos y la información que se muestra al usuario. Este es uno de los elementos más importantes de una aplicación móvil debido a que es el que define la experiencia del usuario. Cómo interactúa el usuario con la aplicación será la diferencia entre una aplicación exitosa y los demás.

2.2.2. Lógica de la aplicación e intrfaces

Lógica de la aplicación es la parte del programa que define el comportamiento de la aplicación. Se determinará qué acciones se ejecutarán en respuesta a diferentes tipos de estímulos como la interacción del usuario, las llamadas telefónicas, los programadores internos, etc Esta parte va a interactuar con el software instalado en el sistema operativo

para poder acceder a diversas funciones como la cámara , Internet, la agenda y todas las características de otro tipo que el sistema operativo del dispositivo permite el acceso. Para el alcance de esta tesis, a las siguientes características que Android y iOS proporcionen serán estudiados: cámara, audio, GPS, mapas y almacenamiento.

2.3. Cross-platform application

Una aplicación multi-plataforma es un software desarrollado para funcionar en diferentes plataformas sin conducta o cambios visuales. Hay varios niveles de plataformas cruzadas aplicaciones. En un lado de la balanza, están los completos aplicaciones multiplataforma como una aplicación de Java, donde no se necesitan cambios en el código para poder funcionar sobre cualquier plataforma de forma como MacOS, Windows o Linux. En el medio, hay aplicaciones que necesitan pequeñas modificaciones en el interior del código que se ejecuta en plataformas diferentes, pero casi toda la codificación es la misma. Un ejemplo podría ser una aplicación web escrita en Hyper-Text Markup Language (HTML), Cascading Style Sheets (CSS) y JavaScript. Si bien estos son los estándares del Consorcio World Wide Web [17], todos los navegadores web diferente que implemente de una manera diferente. Así, una modificación de la codificación es necesaria con el fin de tener una plataforma cruzada (en este caso, una cruz-explorador) aplicación. En el lado opuesto de la escala, están las aplicaciones que están diseñadas una vez y pro-programada varias veces, una aplicación por plataforma. Con esta técnica una aplicación se crea, que no es realmente multiplataforma, pero para el usuario final que parece. Cuando se trata de aplicaciones multiplataforma móviles, el tipo más común son el segundo y el tercer tipo. El segundo tipo, la puesta en práctica de una sola vez, más las modificaciones por la plataforma, es utilizado principalmente por las aplicaciones basadas en web. En el otro lado, el tercer tipo está totalmente ligado con las aplicaciones nativas.

3. APPCELERATOR TITANIUM

Titanium Appcelerator[18] es una plataforma de código abierto que proporciona las herramientas para desarrollar aplicaciones multiplataforma para dispositivos móviles utilizando JavaScript [22] como la tecnología central. Titanium permite a los usuarios programar aplicaciones basadas en Web para nativos y plataformas iOS y Android..

Para interactuar con Titanium Appcelerator, el marco Titanium Studio [23] se requiere. Este marco se basa en el IDE de Eclipse [24] por lo que permite a los usuarios a hacer las mismas cosas que pueden hacer con el proyecto Eclipse original, es decir, codificación, prueba, implementación y control de versiones. También incluye la posibilidad de instalar la aplicación directamente en el dispositivo para probarlo y depurarlo. Esto es particularmente interesante porque hay características como la cámara, la grabadora de audio o el acelerómetro que no son accesibles con las emulaciones tores que XCode o SDK de Android proporciona..

Titanium actúa como un puente entre las diferentes implementaciones de los componentes de una aplicación de telefonía móvil realizadas por los diferentes sistemas operativos (Android y iOS) y el código de programación JavaScript. El diagrama correspondiente se muestra en el lado derecho de la Figura 3,1.

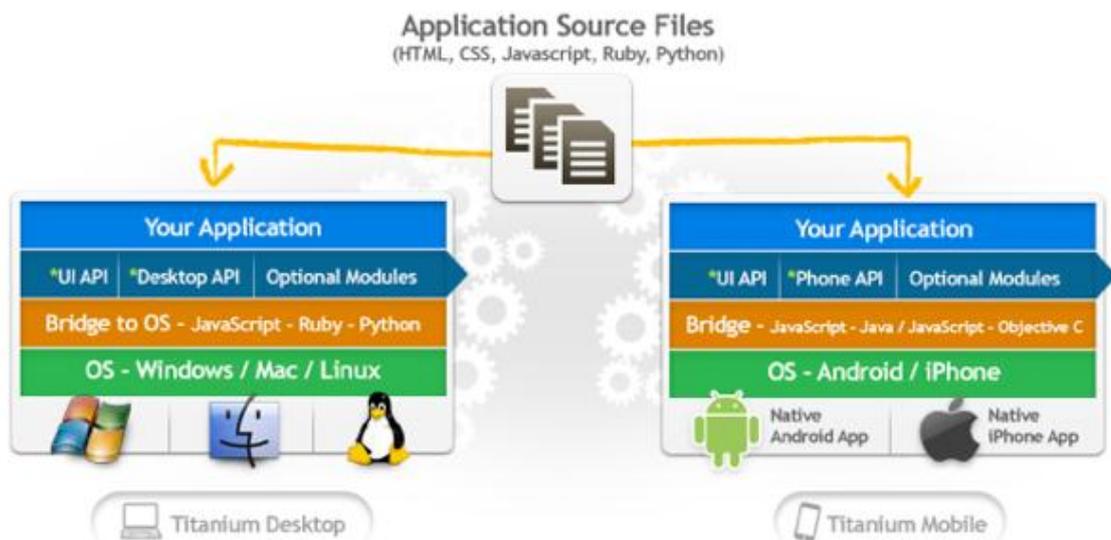


Figure 3.1 Titanium Mobile architecture

4. ANÁLISIS DE UNA APLICACIÓN MÓVIL MULTI PLATAFORMA

Una vez que sabemos cómo usar las herramientas que Appcelerator ofrece, podemos crear una aplicación que haga uso de estas. En este capítulo vamos a crear un aplicación de ejemplo con la que podremos analizar cómo se integran todas estas características en una misma aplicación. Como fue mencionado en el capítulo anterior, las herramientas que van a ser usadas para desarrollar la aplicación son un ordenador iMac con el sistema operativo MacOS 10.X, el framework de compilación Xcode 4.2.1, el SDK 3.1 de Android y el framework Appcelerator Titanium 1.8.1. La aplicación será testeada en un MOTOROLA Xoom Tablet en el caso de Android, y en simulador del iPhone para iOS.

Al final de este capítulo, se realizará un análisis del proceso de desarrollo, puntualizando las dificultades y diferencias encontradas a la hora de integrar todas las características especificadas en el capítulo anterior en el mismo sistema.

4.1. Caso de uso: Aplicación para crear guías de viaje

El caso de uso que nos servirá para analizar el framework es una aplicación basada en la geo localización del dispositivo móvil. El objetivo principal de esta aplicación es ofrecer al usuario la posibilidad de crear sus propias guías de viaje digitales mientras va caminando por la calle. Una guía de viaje digital es una versión simplificada de la versión física de un libro guía de viaje.

Cuando un usuario crea una guía, tiene que introducir un nombre y una descripción, y tiene que especificar la ciudad a la que la guía pertenece. Además, el usuario puede añadir lugares de interés a la guía. Cuando el usuario se encuentra en un lugar que considera interesante, puede añadirlo a la guía y en ese instante se creará una posición relativa a la localización GPS del usuario. Un lugar está definido por un nombre y una descripción, y además el usuario puede añadir fotos obtenidas con la cámara del dispositivo o audio grabado con el micrófono. Basándonos en esta descripción, la aplicación puede hacer lo siguiente:

1. Crear una nueva guía de viaje. Una guía está definida por un nombre y una ciudad (campos obligatorios) y una descripción.
2. Crear un nuevo lugar dentro de la guía. Un lugar está definido por un nombre (campo obligatorio), las coordenadas (obtenidas automáticamente por la aplicación)

y una descripción. El usuario además puede asociar el punto con una fotografía tomada por la cámara y un audio tomado por el micrófono.

3. Visualizar las guías creadas. La aplicación debe mostrar una lista con todas las guías y ofrecer la posibilidad de acceder a la información de dichas guías.
4. Visualizar los lugares creados dentro de una determinada guía. La aplicación debe mostrar una lista con todos los lugares pertenecientes a una determinada guía y ofrecer la posibilidad de acceder a la información de dicho lugar..
5. Editar la información de una guía ya creada. El usuario podrá modificar el nombre, la ciudad y la descripción de la guía. La aplicación también ofrecerá la posibilidad de eliminar, editar o crear nuevos lugares sobre una guía ya creada.
6. La información de un lugar ya creado. El usuario podrá modificar le nombre y la descripción del lugar pero no podrá ni editar, ni eliminar, ni añadir fotos a un lugar ya creado.
7. Eliminar una guía. La aplicación eliminará la guía tanto de la lista de guías creadas como de todas las estructuras internas donde esté almacenada..
8. Mostrar todos los lugares pertenecientes a una guía dentro de un mapa. Para esta característica el usuario necesitará tener conexión a Internet en el momento de la visualización.

Además, la aplicación tiene las siguientes especificaciones técnicas:

1. El dispositivo donde se instale la aplicación necesitará contar con un sistema de geo localización, ya sea por GPS o localización vía Wi-Fi.
2. La aplicación está desarrollada para ser usada en dispositivos con la versión 3.1 de Android o superiores.
3. La aplicación está desarrollada para ser usada en dispositivos iPhone con la versión 4.X de iOS.

4.2. Análisis de los requisitos

Una vez se conocen los requisitos de la aplicación, se puede realizar un análisis de los mismos y extraer la información necesaria para implementar la funcionalidad requerida para construir el sistema.

Crear una nueva guía (**Figure 4.1**) consiste en la creación de un objeto que contiene la información básica de una guía y punteros a los lugares que pertenecen a dicha guía.

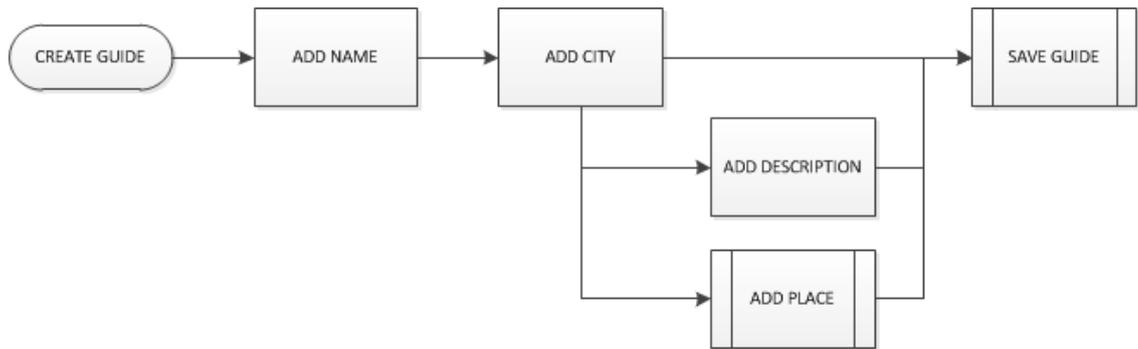


Figure 4.1 Crear guía

Crear un nuevo lugar (**Figure 4.2**) consiste en la creación de un objeto que contendrá la información básica de dicho lugar. Cada lugar tendrá un número de identificación que servirá para asociar los posibles ficheros de imagen y audio que contenga ese lugar.

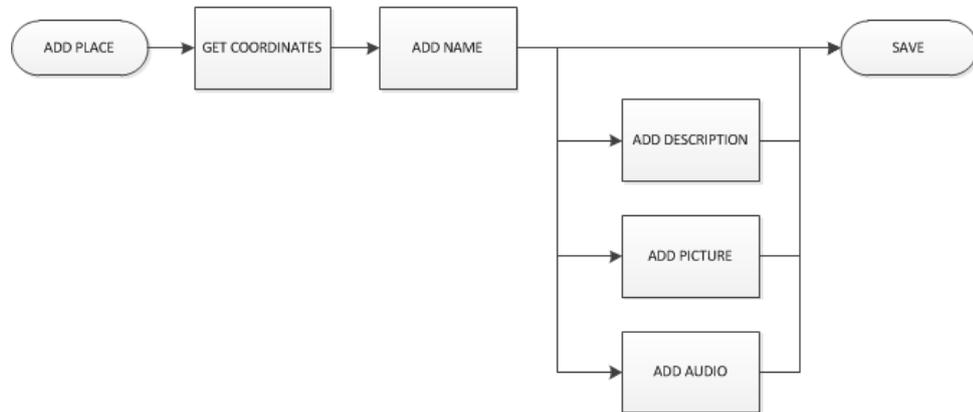


Figure 4.2 Crear lugar

Guardar una guía (**Figure 4.3**) consiste en la creación de una carpeta dentro del sistema de ficheros perteneciente a la aplicación. Esta carpeta tendrá por nombre el número de identificación de la guía. Dentro de esta carpeta se creará un fichero XML, *info.xml*, que contendrá la información básica: El número de identificación (dado por la aplicación), el nombre, la ciudad y la descripción (opcional). Además, dentro de esta carpeta también habrá carpetas destinadas a los lugares pertenecientes a la guía. El nombre de estas carpetas seguirá el mismo criterio que su padre y usarán un identificador (en este caso el identificador del lugar) como nombre. En cada carpeta de lugar se incluirá un fichero XML, *info.xml*, que contiene la información básica del lugar: el número de identificación (dado por la aplicación), el nombre, la descripción (opcional), la latitud y la longitud, y una referencia al lugar donde los archivos de foto y audio del lugar se deben guardar durante el proceso de creación de la guía.

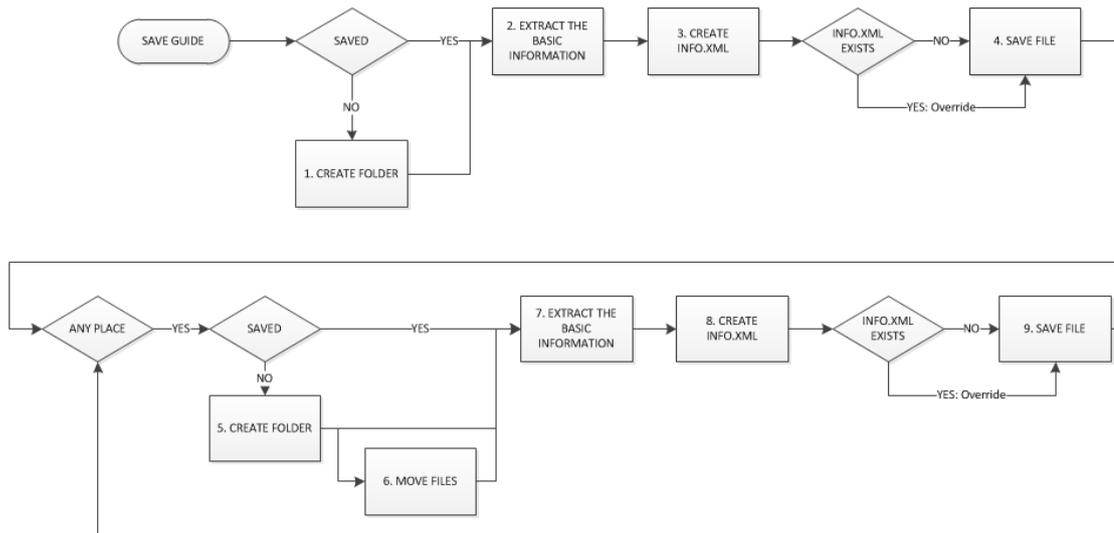


Figure 4.3 Guardar guía

4.3. Diseño de la aplicación

Una vez que el comportamiento de la aplicación está definido, tenemos que diseñar la aplicación y traducir los diagramas de diseño en código JavaScript. La estructura de la aplicación está definida en la **Figure 4.4**.

- El directorio de la aplicación contiene una carpeta llamada *Resources* con el código de la misma.
- El directorio *PLACE_AUX* contiene la estructura de ficheros necesaria para almacenar temporalmente los archivos de fotos y audio generados durante el proceso de creación de lugares.
- El directorio *MY_GUIDES* contiene la estructura de ficheros necesaria para almacenar toda la información de las guías creadas por el usuario. Este directorio se mantiene hasta que la aplicación es borrada del dispositivo.

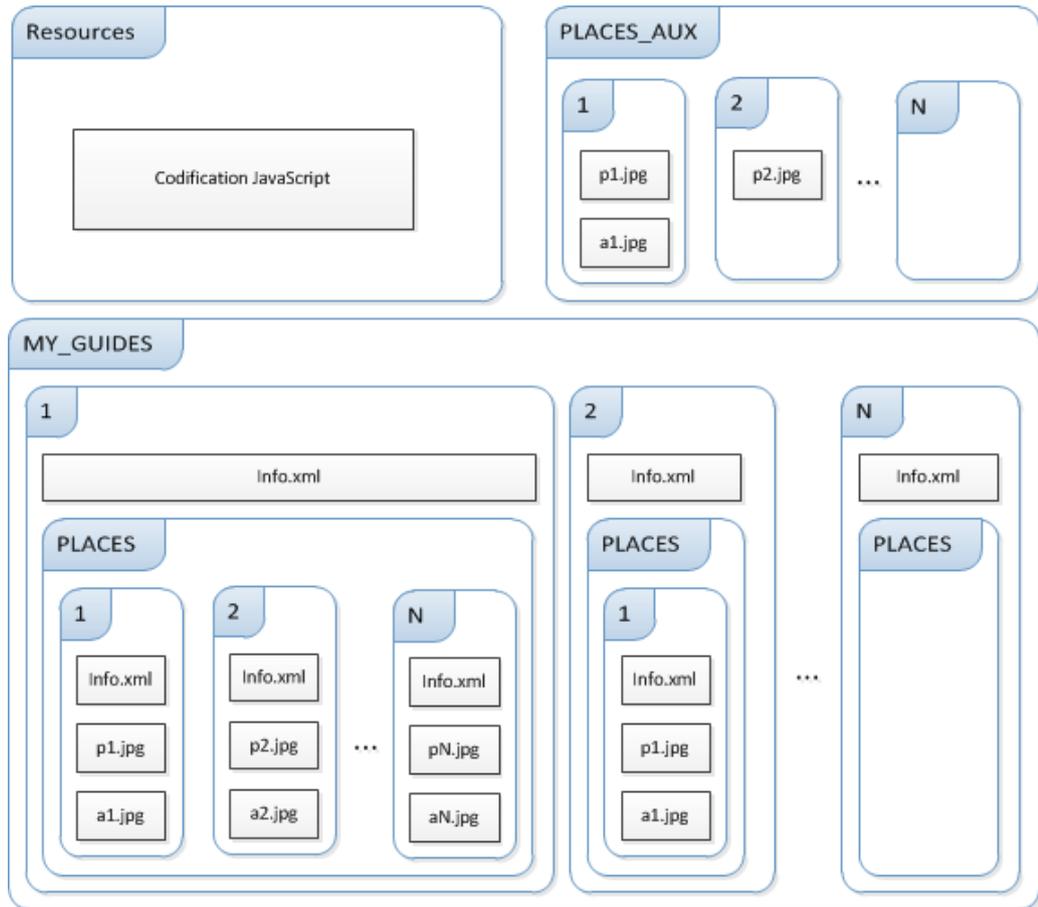


Figure 4.4 Estructura de la aplicación

La implementación de la aplicación está estructurada según el diseño mostrado en la **Figure 4.5**.

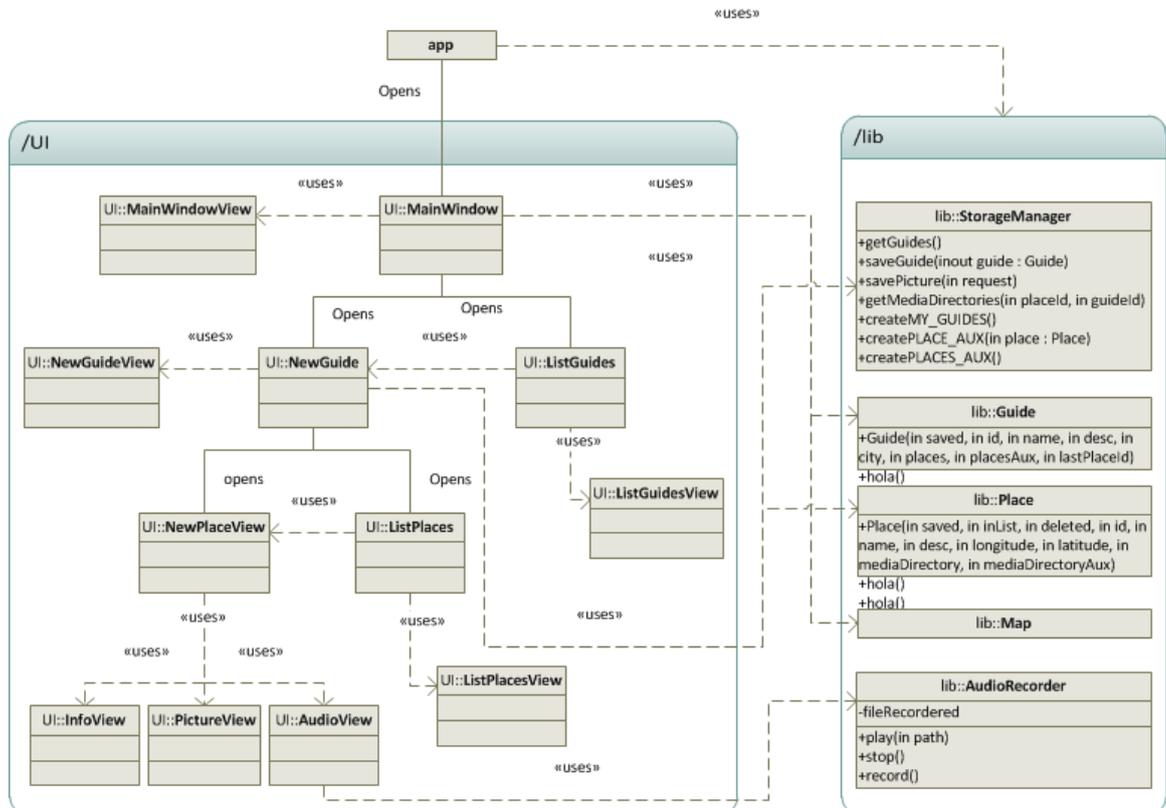


Figure 4.5 Diagrama de la aplicación.

La aplicación está dividida en tres módulos principales: el módulo de la interfaz de usuario (UI), el módulo para las librerías (lib) y el módulo con la funcionalidad principal (app). Estos módulos serán explicados a continuación.

4.3.1. Módulo UI

El módulo UI es el encargado de albergar la funcionalidad que servirá para mostrar la información de la aplicación al usuario.

4.3.2. Módulo lib

El módulo lib encapsula la funcionalidad externa a la aplicación (interactuar con el sistema de ficheros, mapas, etc).

4.3.3. Módulo App

Este módulo contiene únicamente un elemento, el *app.js*. Este objeto contiene el punto de ejecución inicial de la aplicación y es el responsable de inicializar las interfaces gráficas.

ficas. También es el responsable de escuchar las posibles peticiones que los demás módulos puedan solicitar (guardar guía, leer guías, etc)..

4.3.4. Pantallazos

En la **Figure 4.6** podemos ver la ventana principal de la aplicación donde el usuario puede crear una nueva guía o acceder a la lista de guías que fueron creadas.

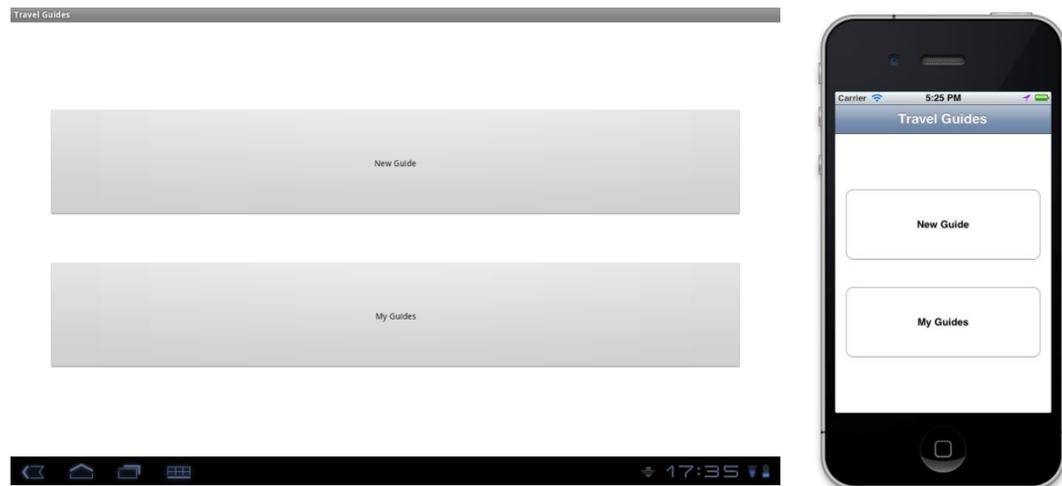


Figure 4.6 Main Window en Android (izquierda) y iOS (derecha)

En la **Figure 4.7** podemos ver la interfaz para definir la información de una nueva guía.

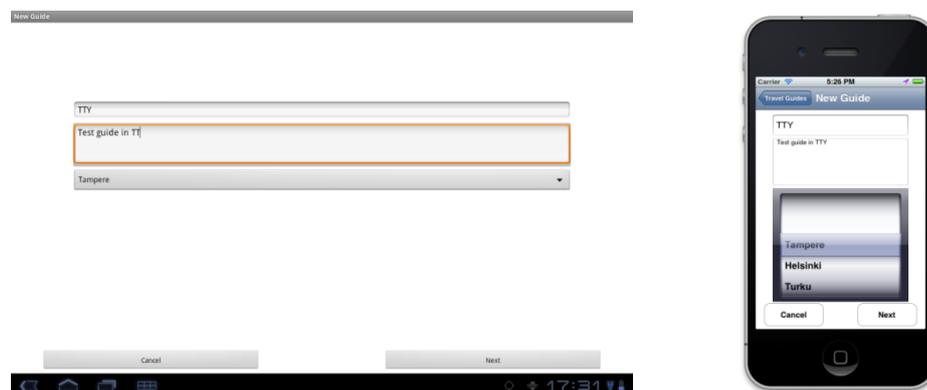


Figure 4.7 New Guide: Información básica en Android (izquierda) y iOS (derecha)

En la **Figure 4.8** podemos ver la interfaz donde el usuario puede crear un nuevo lugar dentro de una guía.

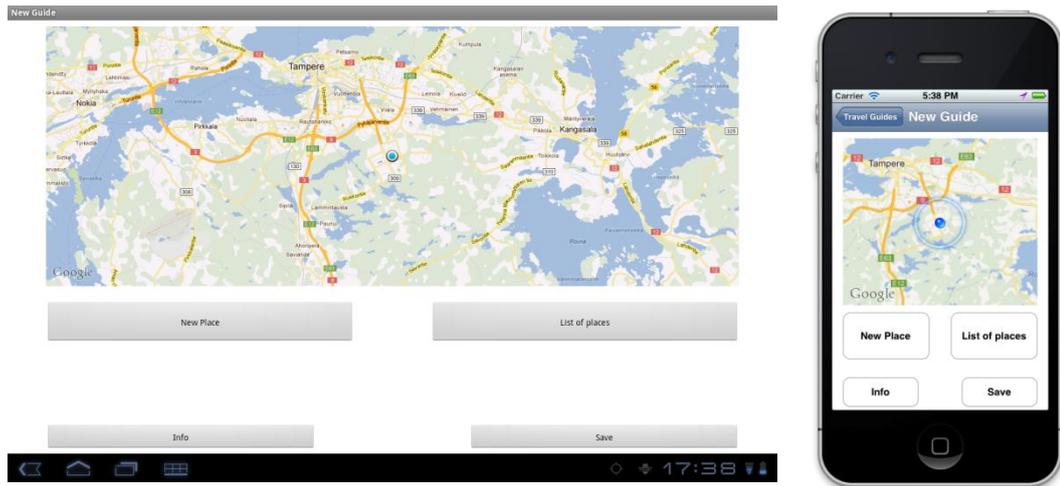


Figure 4.8 *New Guide: Places en Android (izquierda) y iOS (derecha)*

En la **Figure 4.9** podemos ver la información básica para la creación de un nuevo lugar.

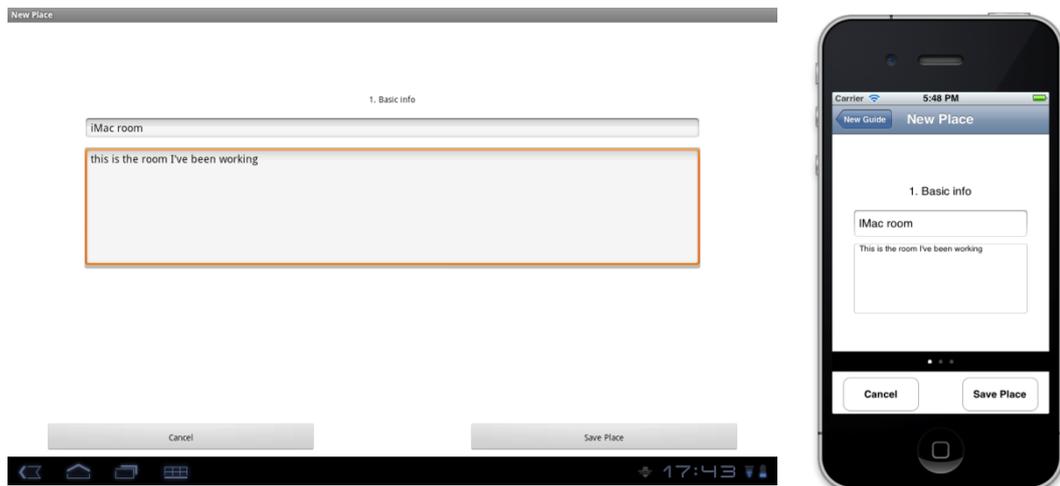


Figure 4.9 *New Place: Basic Info en Android (izquierda) y iOS (derecha)*

En la **Figure 4.10** podemos ver la segunda interfaz en la creación de un nuevo lugar. Esta interfaz corresponde a la toma de fotos con la cámara del dispositivo.

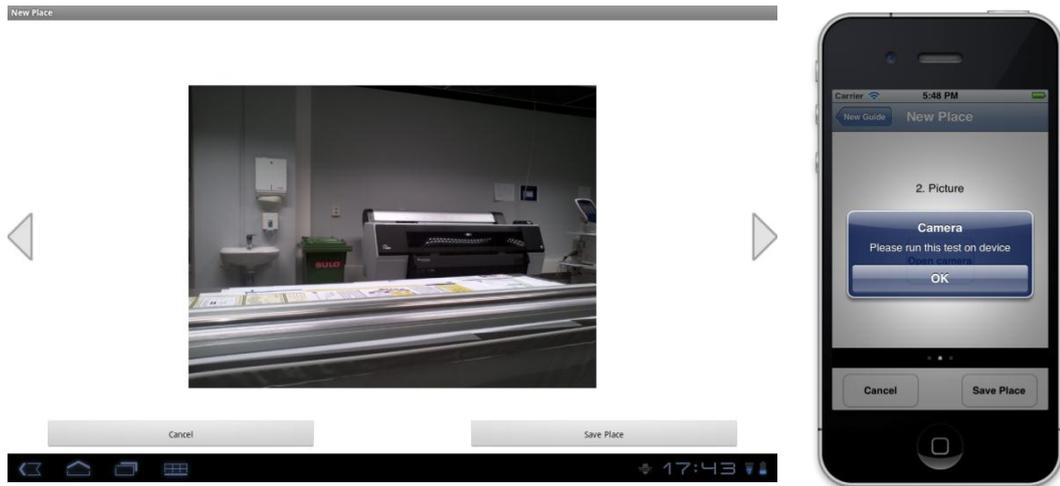


Figure 4.10 *New Place: Picture view after take the picture en Android (izquierda) y iOS (derecha)*

La última interfaz perteneciente al grupo de creación de nuevos lugares se muestra en la **Figure 4.11**. En esta pantalla podemos ver la interfaz para añadir un audio al nuevo lugar.



Figure 4.11 *New Place: Audio en Android (izquierda) y iOS (derecha)*

In **Figure 4.12** we can see how the interface informs the user about a mandatory field. In this case is informing to the user that the input field needed to specify the name of the place is empty.

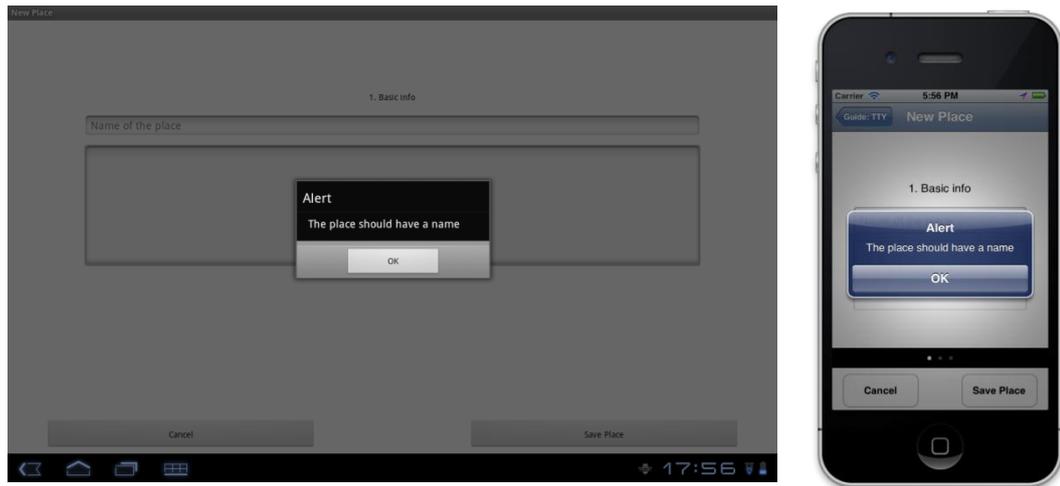


Figure 4.12 *New Place: No name alert in Android (left) and iOS (right)*

In **Figure 4.13** we can see the list of guides of the user. Every element of the list contains two elements: A text with the name of the guide and a button with the “x” character to delete the guide from the application. In order to open a specific guide, the user has to click in the specific row of the list.



Figure 4.13 *List of guides in Android (left) and iOS (right)*

In **Figure 4.14** we can see the map with the places belonging to one guide. As it was explained on a previous figure, the places are shown on the map. A green pin is used in order to point the places on the map. To visualize the basic information of the place the user has to click in any of the pins.

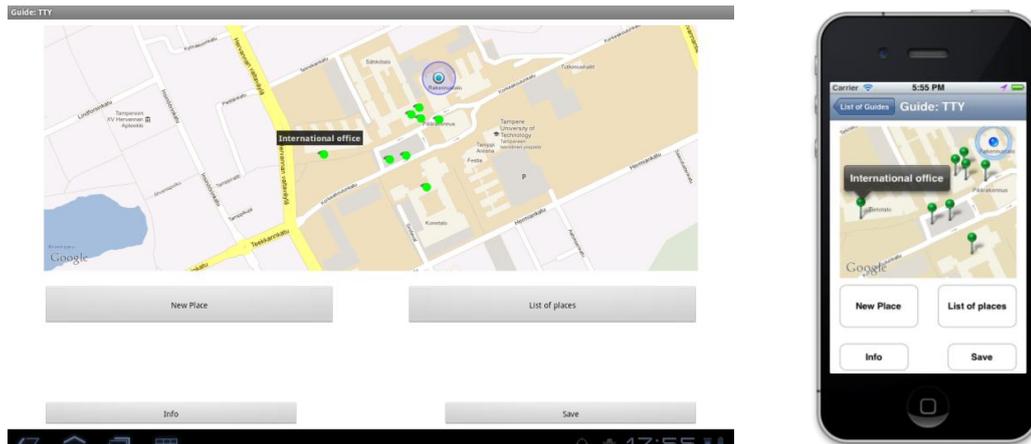


Figure 4.14 List of places: Map in Android (left) and iOS (right)

In **Figure 4.15** we can see the list of places belonging to one guide. Every element of the list contains two elements: A text with the name of the place and a button with the “x” character to delete the place from the guide. In order to open an specific place, the user has to click in the specific row of the list.



Figure 4.15 List of places: List in Android (left) and iOS (right)

4.4. Evaluación del framework

El primer punto a tratar en esta evaluación trata sobre los sistemas operativos donde el framework puede operar. Durante el desarrollo de este proyecto se han usado tanto entornos Linux como Windows y MacOS. En todos ellos se trabaja perfectamente pero en los entornos MacOS existe una ventaja. Mientras que el Android SDK funciona en los tres entornos, el XCode (necesario para compilar aplicaciones para iOS) únicamente funciona en MacOS. Por esta razón es aconsejable usar Appcelerator únicamente en sistemas (físicos o virtuales) con el sistema operativo de Apple.

El segundo punto para ser analizado es cómo es la interacción con el framework, incluida la parte de programación, el soporte y las dependencias externas. En lo referente a la parte de programación, JavaScript es comúnmente usado en entornos Web. Pero en este caso el uso que se le da es un poco diferente al que se usa en la Web. Por esta razón, los usuarios que conocen JavaScript debido al desarrollo en la Web tendrán que adaptarse a los nuevos métodos y técnicas de trabajo. Este factor hace que el proceso de aprendizaje sea lento al principio pero una vez que se alcanza un nivel de conocimiento suficiente el proceso de desarrollo se acelera. Para ayudar en este proceso un *debug mode* está disponible. Este modo sólo está disponible en dispositivos virtuales. Esto puede llegar a ser un problema ya que el emulador de Android que crea los dispositivos virtuales es lento.

Estas pequeñas desventajas no son suficientes para determinar si Appcelerator es una buena opción para desarrollar aplicaciones multi plataforma o no lo es. Por otro lado, existe una característica que definitivamente es un punto determinante a la hora de seleccionar una tecnología o no. Appcelerator no ofrece una herramienta gráfica para desarrollar las interfaces gráficas de las aplicaciones. A pesar de que Appcelerator recomienda a los desarrolladores que usen patrones MVC para separar el comportamiento de la aplicación de la presentación, esto no es suficiente para crear y mantener buenas interfaces gráficas. Cuando los desarrolladores de aplicaciones móviles utilizan frameworks para el proceso de desarrollo normalmente usan herramientas gráficas como las mostradas en la **Figure 4.16**. Esto facilita mucho la tarea de desarrollar y se podría considerar una desventaja para el desarrollo con Appcelerator.

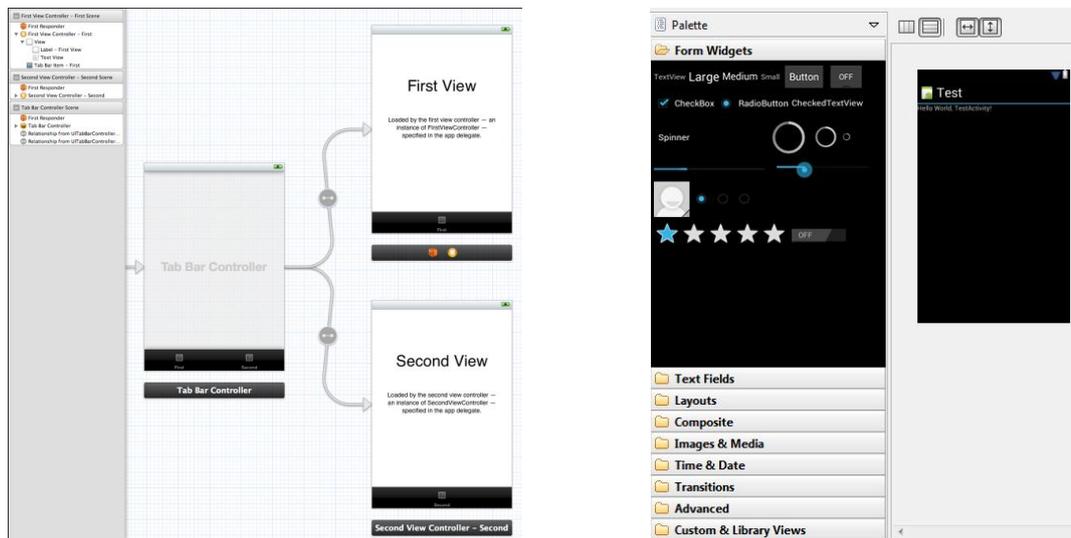


Figure 4.16 Graphical GUI development tool in XCode and Eclipse

Es bastante probable que cualquier desarrollador se pueda encontrar problemas con los frameworks que utilizan, problemas que ellos no puedan resolver por sí solos por lo que necesitan de un soporte y una comunidad online que ayude en estos aspectos. En este

aspecto Appcelerator posee un buen sistema de soporte a través de su página web. Tienen un gestor de errores bueno y hay una gran comunidad que contribuye a la "Preguntas y respuestas" que hizo más fácil resolver problemas. Por otro lado, Appcelerator tiene una larga lista de bugs sin resolver que pueden hacer que un desarrollador se arrepiente de su decisión de utilizar este framework.

Otro punto de vista a tener en cuenta es el nivel de dependencia entre el framework y la nativa SDK (Android o iOS). Hay una clara dependencia entre ellos, pero no afecta directamente el desarrollo. Cuando una aplicación móvil se diseña, las versiones del SDK establecen y una nueva actualización del SDK no afectará al desarrollo de la aplicación, porque está fuera del alcance del proyecto.

Una vez que la aplicación está desarrollada, es hora de empaquetar todos los recursos para poder distribuirla. En esta fase, Appcelerator demuestra una clara falta de transparencia y flexibilidad debido al hecho de que no proporciona el código nativo (Java o ObjectiveC). Para modificar la solución final con el fin de realizar mejoras o cambios futuros que demanda la plataforma nativa no es posible. Appcelerator empaqueta todo en el archivo de la aplicación final (.apk o .ipa.) listo para su distribución. Se podría considerar un defecto o una virtud. Algunos desarrolladores podrían considerar esto algo para agilizar el proceso de publicar la aplicación en los mercados en línea, pero otros podrían ver que es un aspecto negativo debido a la imposibilidad de saber cómo se implementa la aplicación. Pero sin duda es una característica a tener en cuenta antes de empezar a desarrollar una aplicación con esta plataforma.

El último aspecto a tener en cuenta es el grado de similitud entre las aplicaciones producidas para Android y el iOS y el nivel de la programación paralela que se tiene que hacer con el fin de cubrir los aspectos nativos de las plataformas individuales. En esta parte Appcelerator funcionando bastante bien. El nivel de la programación en paralelo no es demasiado alto por lo que la mayor parte del código no tiene por qué ser dividido en dos para cubrir este aspecto. Durante el desarrollo de los ejemplos y la aplicación de casos de uso solo los diferentes sistemas para gestionar la jerarquía de las ventanas y la navegación entre Android y iOS debía gestionarse de forma paralela. El resto del código es compartido entre plataformas. Es cierto que algunos de los elementos gráficos que se pueden agregar a las interfaces de usuario como botones o recolectores son un poco diferentes, pero estas diferencias no son tan visibles cuando se trata de pequeñas aplicaciones sin interfaces de usuario muy elegantes.

5. CONCLUSIONS

En esta tesis se analiza el desarrollo de aplicaciones multi-plataforma móvil para saber las virtudes y defectos de este tipo de aplicaciones. Para este propósito el marco Titanium Appcelerator fue elegido. Uso de Appcelerator una aplicación móvil para Android y el IOS fue creado. El desarrollo de esta aplicación se utiliza para analizar el marco y el proceso de desarrollo.

Las conclusiones generales extraídas del desarrollo de esta tesis y la aplicación es que las aplicaciones multi-plataforma móvil tiene muchas ventajas comparando con las aplicaciones móviles regulares, pero el proceso de desarrollo depende de los marcos que no son lo suficientemente maduros.

Hay varias soluciones en el mercado que ayuda en la creación de aplicaciones multiplataforma móviles. Cada uno de ellos utiliza diferentes técnicas y conducen a resultados diferentes, pero la característica común de todos ellos es que utilizan tecnologías web para crear las aplicaciones.

Titanium Appcelerator es un framework fácil de utilizar que tiene una gran comunidad de desarrolladores. Las soluciones creadas con este marco, matienen un alto nivel de similitud. Pequeños cambios en el código son necesarias para el desarrollo de aplicaciones que tienen el mismo comportamiento y el aspecto de los dispositivos Android y iOS.

El proceso de aprendizaje con Appcelerator es lento en el comienzo. Uno puede encontrar muchos problemas, mientras se comenza a utilizar el marco, pero una vez que sabemos cómo usarlo, el tiempo de desarrollo de una aplicación multiplataforma se reduce.

A pesar de que Appcelerator parecen tener características que lo hacen apto para desarrollar aplicaciones multiplataforma móviles, todavía tiene tres desventajas importantes. El primero es la estabilidad. Durante el desarrollo de esta tesis Appcelerator actualizado cuatro veces la interfaz de programación de aplicaciones (API). Algunas de las mejoras no afectan al proceso de desarrollo, pero sí los demás. La desventaja siguiente es el desarrollo de las interfaces gráficas de usuario. El desarrollo de interfaces gráficas de usuario que utilizan los recursos nativos se hace de forma gráfica. La mayoría de los marcos, como Eclipse o XCode proporcionar herramientas para simplificar este proceso.

Con Appcelerator este proceso se hace escribiendo el código que representa a las interfaces. Este hecho hizo que el desarrollo lento e impreciso. El último inconveniente es la gran cantidad de errores que aún están abiertas. Para encontrar uno de estos errores es algo que todos los desarrolladores prefiere evitar.

Por último, desde el punto de vista de los desarrolladores y empresas, para crear aplicaciones multiplataforma es algo deseable, ya que pueden llegar a los compradores más posibles pero hoy en día sigue siendo más rentable para crear aplicaciones diferentes en el mismo proceso de desarrollo en lugar de utilizar cualquiera de estas las nuevas tecnologías.

Para resumir todo, aquí en la **Table 1** los resultados extraídos de esta tesis se muestran. Estos resultados están relacionados con Appcelerator.

Table 1 Titanium Appcelerator analysis results

Creación de interfaces	2	Pobre creación y modificación de interfaces
Facilidad de programación	4	Las técnicas de programación no son muy difíciles de aprender
Proceso de aprendizaje	3	Lento al principio pero más rápido después
Soporte de la comunidad	4	Gran comunidad de desarrollo y soporte

1: *Malo.*

2: *No es muy malo pero necesita mejoras.*

3: *Bueno.*

4: *Muy bueno pero no tan bueno como el desarrollo nativo.*

5: *Tan bueno como el desarrollo nativo.*